

Claims

What is claimed is:

1. 1. A method comprising:
 2. in a processor based system where a plurality of processors share processor execution resources, in response to a first processor in the plurality of processors being scheduled to enter an idle state, making a processor execution resource previously reserved for the first processor available to a second processor in the plurality of processors.
 1. 2. The method of claim 1 further comprising reserving the processor execution resource for the first processor in response to the first processor being scheduled to execute a task.
 1. 3. The method of claim 2 wherein each of the plurality of processors is a logical processor of the processor based system.
 1. 4. The method of claim 3 wherein the first processor being scheduled to enter an idle state further comprises the first processor executing a processor instruction requesting the first processor to enter an idle state.
 1. 5. The method of claim 4 wherein making the processor execution resource previously reserved for the first processor available to a second processor further comprises releasing the processor execution resource into a common pool of processor execution resources accessible from the second processor.

1 6. The method of claim 5 wherein the first processor being scheduled to execute a task
2 further comprises the first processor receiving a wake up signal.

1 7. The method of claim 6 wherein the processor execution resource previously reserved
2 for the first processor further comprises the processor execution resource previously
3 statically allocated to the first processor; and wherein releasing the processor
4 execution resource into a common pool of processor execution resources further
5 comprises de-allocating the processor execution resource.

1 8. The method of claim 6 wherein the processor execution resource previously reserved
2 for the first processor further comprises the processor execution resource previously
3 locked by the first processor; and wherein releasing the processor execution resource
4 into a common pool of processor execution resources further comprises the first
5 processor unlocking the processor execution resource.

1 9. The method of claim 6 wherein the common pool of processor execution resources
2 comprises a translation lookaside buffer and the processor execution resource is a
3 translation cache entry from the translation lookaside buffer.

1 10. A processor comprising:
2 a plurality of logical processors; and
3 an instruction set, the instruction set comprising one or more instructions which when
4 executed by a first logical processor, cause the first logical processor to make a
5 processor execution resource previously reserved for the first processor available to a
6 second processor in the plurality of processors in response to the first logical
7 processor being scheduled to enter an idle state.

- 1 11. The processor of claim 10 wherein to the first logical processor being scheduled to
- 2 enter an idle state further comprises the first processor executing a processor
- 3 instruction requesting the first logical processor to enter an idle state.
- 1 12. The processor of claim 11 wherein causing the first logical processor to make the
- 2 processor execution resource previously reserved for the first logical processor
- 3 available to a second logical processor further comprises releasing the processor
- 4 execution resource into a common pool of processor execution resources accessible
- 5 from the second logical processor.
- 1 13. The processor of claim 12 wherein the processor execution resource previously
- 2 reserved for the first logical processor further comprises the processor execution
- 3 resource previously statically allocated to the first logical processor; and wherein
- 4 releasing the processor execution resource into a common pool of processor
- 5 execution resources further comprises de-allocating the processor execution resource.
- 1 14. The processor of claim 12 wherein the processor execution resource previously
- 2 reserved for the first logical processor further comprises the processor execution
- 3 resource previously statically allocated to the first logical processor; and wherein
- 4 releasing the processor execution resource into a common pool of processor
- 5 execution resources further comprises the first processor unlocking the processor
- 6 execution resource.
- 1 15. A system comprising:
 - 2 a processor, the processor comprising
 - 3 a plurality of logical processors; and

4 an instruction set, the instruction set comprising one or more instructions
5 which when executed by a first logical processor, cause the first logical
6 processor to make a processor execution resource previously reserved for
7 the first processor available to a second processor in the plurality of
8 processors in response to the first logical processor being scheduled to
9 enter an idle state;
10 firmware to schedule the first logical processor to enter an idle state; and
11 a bus to interconnect the firmware and the processor.

1 16. The system of claim 15 wherein the first logical processor being scheduled to enter an
2 idle state further comprises the first processor executing a processor instruction
3 requesting the first logical processor to enter an idle state.

1 17. The system of claim 16 wherein causing the first logical processor to make the
2 processor execution resource previously reserved for the first logical processor
3 available to a second logical processor further comprises releasing the processor
4 execution resource into a common pool of processor execution resources accessible
5 from the second logical processor.

1 18. The system of claim 17 wherein the processor execution resource previously
2 reserved for the first logical processor further comprises the processor execution
3 resource previously statically allocated to the first logical processor; and wherein
4 releasing the processor execution resource into a common pool of processor
5 execution resources further comprises de-allocating the processor execution resource

1 19. The system of claim 17 wherein the processor execution resource previously
2 reserved for the first logical processor further comprises the processor execution
3 resource previously statically allocated to the first logical processor; and wherein
4 releasing the processor execution resource into a common pool of processor
5 execution resources further comprises the first processor unlocking the processor
6 execution resource.

1 20. A machine accessible medium having stored thereon data which when accessed by a
2 machine causes the machine to perform a method, the method comprising:
3 in a processor based system where a plurality of processors share processor execution
4 resources, in response to a first processor in the plurality of processors being
5 scheduled to enter an idle state, making a processor execution resource previously
6 reserved for the first processor available to a second processor in the plurality of
7 processors.

1 21. The machine accessible medium of claim 20 further comprising reserving the
2 processor execution resource for the first processor in response to the first processor
3 being scheduled to execute a task.

1 22. The machine accessible medium of claim 21 wherein each of the plurality of
2 processors is a logical processor of the processor based system.

1 23. The machine accessible medium of claim 22 wherein the first processor being
2 scheduled to enter an idle state further comprises the first processor executing a
3 processor instruction requesting the first processor to enter an idle state.

- 1 24. The machine accessible medium of claim 23 wherein making the processor execution
2 resource previously reserved for the first processor available to a second processor
3 further comprises releasing the processor execution resource into a common pool of
4 processor execution resources accessible from the second processor.
- 1 25. The machine accessible medium of claim 24 wherein the first processor being
2 scheduled to execute a task further comprises the first processor receiving a wake up
3 signal.
- 1 26. The machine accessible medium of claim 25 wherein the processor execution
2 resource previously reserved for the first processor further comprises the processor
3 execution resource previously statically allocated to the first processor; and wherein
4 releasing the processor execution resource into a common pool of processor
5 execution resources further comprises de-allocating the processor execution resource.
- 1 27. The machine accessible medium of claim 25 wherein the processor execution
2 resource previously reserved for the first processor further comprises the processor
3 execution resource previously locked by the first processor; and wherein releasing the
4 processor execution resource into a common pool of processor execution resources
5 further comprises the first processor unlocking the processor execution resource.
- 1 28. The machine accessible medium of claim 25 wherein the common pool of processor
2 execution resources comprises a translation lookaside buffer and the processor
3 execution resource is a translation cache entry from the translation lookaside buffer.